

Control of Towing Kites for Seagoing Vessels

Michael Erhard and Hans Strauch

Abstract—In this paper we present the basic features of the flight control of the SkySails towing kite system. After introduction of coordinate definitions and basic system dynamics we introduce our model used for controller design and justify its main dynamics with results from system identification based on numerous sea trials. We then present the controller design which we successfully use for operational flights for several years. Finally we explain the generation of dynamical flight patterns.

Index Terms—Aerospace control, Attitude control, Feedforward systems, Wind energy

I. INTRODUCTION

THE SkySails system is a towing kite system which allows modern cargo ships to use the wind as source of power in order to save fuel and therefore to save costs and reduce emissions [1]. The SkySails company has been founded in 2001 and as main business offers wind propulsion systems for ships. Starting the development with kites of 6–10 m² size the latest product generation with a nominal size of 320 m² can replace up to 2 MW of the main engine's propulsion power. Among the marine application of kites there is a strongly increasing activity in using automatically controlled kites [2], [3], [4], [5], [6], [7], [8] and rigid wings [9], [10] in order to generate power from high-altitude wind [11]. Since 2011 the company's second business segment SkySails Power also develops and markets systems for generating power from high-altitude wind. Therefore the design of control systems for kites in these geometries has become a growing field of theoretical [12], [13], [14], [15], [16], [17], [18] and experimental [19], [20], [21] research efforts.

The main components of the SkySails system are shown in Fig. 1. The core of the propulsion system is the towing kite steered by the control pod situated under the kite. The towing force is transmitted to the ship by a high-strength synthetic fiber rope. Additionally a launch and recovery system is installed aboard the ship [1]. One key component of the flight control system is the main steering actuator in the control pod applying deflections to some kite lines leading to curve flight. A control system consisting of distributed computers preprocesses data from various sensors at a rate of 10 Hz and performs the flight control algorithm which calculates the steering command applied to the main actuator. An integrated graphical user interface allows for operation of the system by the crew whereas for research and development purposes



Fig. 1. The BBC SkySails with a 320 m² towing kite.

prototyping and testing toolchains can be connected via special interfaces.

The paper is organized as follows: First we introduce the basic system and coordinate definitions. We then focus on the main dynamics and develop a model specially suited for controller design. After justification of the main law of the model with experimental data we present our controller design discussing design considerations and controller performance measurements. We complete the article with the explanation of pattern generation.

II. BASIC SYSTEM AND COORDINATES

In this section we give a mathematical description of the considered system. It is worth mentioning that we deal with a constrained system which shows a completely different dynamics compared to free flying parafoils [22]. The basic system is sketched in Fig. 2. Compared to the real system we make use of some simplifications. The flexible rope is substituted by a rigid rod which also is parallel to the kite yaw axis \vec{e}_{yaw} and all masses are neglected. At first view this seems to be an unrealistic simplification, but the usual mode of operation is the highly dynamical pattern flight leading to line forces big compared to system masses. Therefore inertia effects or free flight situations, where the towing line is no longer stretched out due to gusts or wave induced ship motions, are infrequent. Although consideration of these issues becomes important at a certain point when bringing the system to higher perfection, a detailed description of the solution to these off-nominal situations would go beyond the scope of this paper.

We would like to start with a demonstrative and introductive example to motivate the idea of the chosen coordinate system φ, ϑ and to explain the basic dynamics. Imagine a kite flying

Manuscript submitted February 16, 2012.

We acknowledge funding from the Federal Ministries BMWI and BMBF, LIFE III of the European Commission, City of Hamburg/BWA and Innovationsstiftung Hamburg.

M. Erhard is with SkySails GmbH, Veritaskai 3, D-21079 Hamburg, Germany, e-mail: michael.erhard@skysails.de, <http://www.skysails.de>.

H. Strauch is a consultant to SkySails.

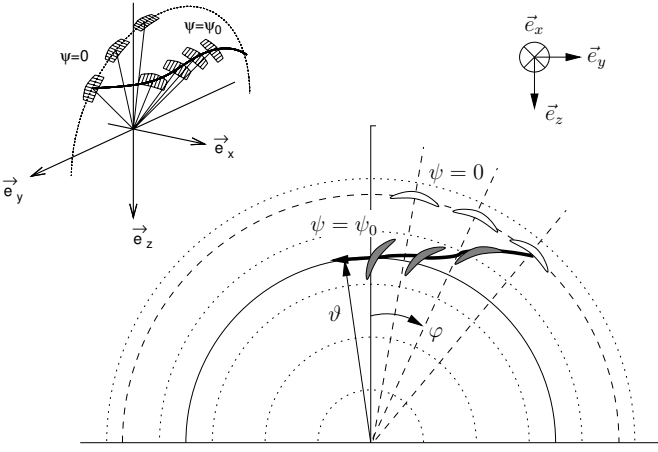


Fig. 4. Zenith positions for neutral flight with low force. With $\psi = 0$ the parameter φ can be freely chosen determining the positions (drawn as white kites). For constant ψ the resulting ϑ converges as shown by the gray kites flying on the marked trajectory.

be reasoned in the next section we neglect acceleration and gravity effects in order to obtain these simple equations of motion which allow for the following interpretations.

First we get a steady state angle ϑ_0 for constant ψ by

$$\vartheta_0(\psi) = \arctan(E \cos \psi). \quad (5)$$

A further result is the dependence of the (steady state) air path velocity v_a on the value of ψ and the ambient wind speed v_0 ,

$$v_a = v_0 E \cos \vartheta = v_0 E \cos(\arctan(E \cos \psi)) \quad (6)$$

which is the key issue for pattern generation as we can use ψ as a tuning knob to control v_a and the force which is proportional to v_a^2 accordingly. It is worth mentioning that for practicing the sport of kite surfing, the content of (6) is crucial in order to control forces: kite surfers know the angle ϑ as position in the so called 'wind window' and the deeper they fly their kite into this 'wind window' — i.e. they decrease this angle — the more traction force they get and vice versa.

A special case is the so called zenith position for $\psi = 0$, a neutral flight situation which generates only low forces and thus is used mainly for launching and recovering the kite. From (5) we get $\vartheta_0 = \arctan(E)$ and φ can be used as free parameter to determine the neutral flight position (compare Fig. 4).

While all equations up to here follow straightforward from kinematic reasoning, albeit the choice of the coordinate system was not 'typical', the dynamic response of the kite to a steering deflection δ is claimed to be

$$\dot{\psi}_m = g v_a \delta. \quad (7)$$

where g is the proportionality factor. We would like to draw special attention to this turn rate law (7) and will show in the following that it can be justified by measured data to a surprising high degree. Therefore it is a key issue for the cascaded controller approach where it constitutes the dynamics of the inner loop.

Finally we would like to point out that due to its motion on a spherical surface an inertial sensor measures a turn rate

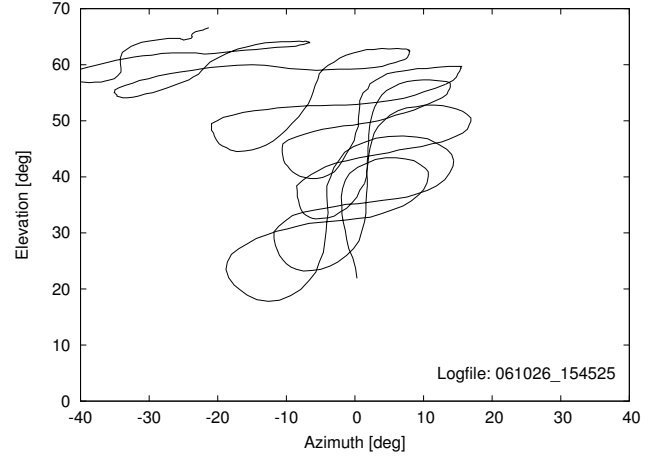


Fig. 5. Flight trajectory under computer control for the bang-bang flight experiment. Angles were measured by tow point sensors on the ship determining the direction of the towing rope. Note that azimuth and elevation values correspond to a different coordinate system which is not defined in this paper, but still show up the figure-eight shape clearly.

$\dot{\psi}_m$ about the yaw axis \vec{e}_{yaw} different from the derivation $\dot{\psi} = d\psi/dt$. The rotation measured by the pod sensors can be calculated by transforming the dynamics represented by \vec{R} into the pod coordinate system by applying (2). Comparing the rotation operation $\vec{\Omega} \times$ with $\vec{R} \cdot \vec{R}^T$ yields:

$$\dot{\psi}_m = \dot{\psi} - \dot{\varphi} \cos \vartheta. \quad (8)$$

By consideration of flight trajectories one can assure oneself that the second term of this equation usually is small compared to the first and thus can be neglected in the first instance and treated as a correction to the controller design later.

IV. JUSTIFICATION OF THE TURN RATE LAW

In this section a justification of (7) is given based on numerous experiments showing the strong proportionality. It is worth noting that (7) is confirmed by measurements to a high degree even in disturbed sea trial conditions. The key issue of these experiments is to perform bang-bang flights which will be presented and discussed in the following.

The excitation of the system for the identification is performed in the following way: We apply a constant steering command $+\delta_0$ to the system. The system will respond with a positive yaw rate ($\dot{\psi}_m > 0$). When reaching a certain threshold $\psi_m \geq \psi_0$ the corresponding opposite steering deflection $-\delta_0$ is commanded leading to a decrease of ψ_m . Falling below the negative threshold $\psi_m \leq -\psi_0$, the primary deflection δ_0 is commanded again. The schedule of the bang-bang experiments is as follows: the human pilot flies the kite into a high zenith position with $\varphi \approx 0$ and hands over the steering to the computer based control system which performs the described algorithm.

A typical flight trajectory is shown in Fig. 5. The bang-bang steering leads to a figure-eight-pattern and for typical parameters the air path speed and thus the size of the pattern increase because the kite flies down to smaller elevation angles ϑ . The human pilot only has the task to supervise the flight and overtake manual control before the system runs into the danger of overload or bounces against the water surface.

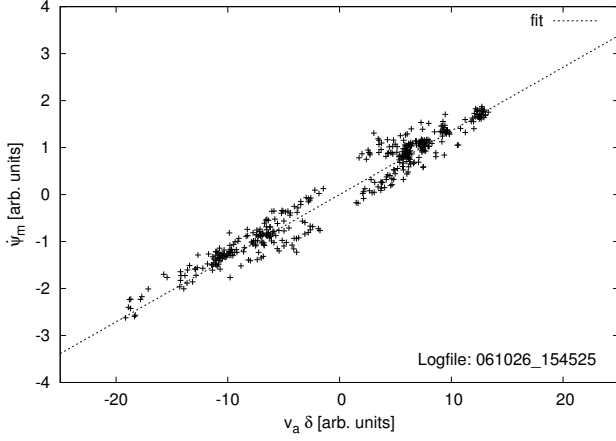


Fig. 6. Measured data of a bang-bang flight. Yaw rate $\dot{\psi}_m$ as function of the air path speed multiplied by deflection $v_a \delta$ to justify (7). The parameter g is obtained by the shown linear fit.

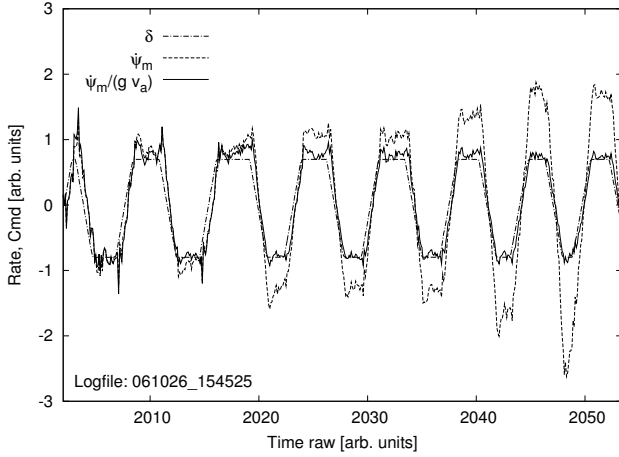


Fig. 7. Comparison of steering command δ with yaw rate $\dot{\psi}_m$ and the yaw rate divided by the air path velocity $\dot{\psi}_m / (g v_a)$. Note the increasing rates are due to increasing air path speed v_a while going down into the wind window (compare Fig. 5).

In Fig. 6 data points for one experiment run are shown. This experiment was performed in 2006 using a 20 m² kite. Repeating this experiment with different δ_0 leads to similar g values and thus proves the validity of (7).

Although the linear dependence can be clearly identified in Fig. 6 it is even more convincing to present the data in the time domain as shown in Fig. 7. Here we compare the time-series of the steering command with the turn rate of the kite. The trapezoidal shape is due to the finite steering velocity of the control pod. The resulting measured yaw rate $\dot{\psi}_m$ shows an increase which results from the increasing air path velocity v_a over the experiment. The yaw rate divided by $g v_a$ is also plotted in order to compare it to the steering command. Although a lot of perturbations affect these experiments we observe an excellent correlation. This analysis justifies the validity of (7) to a high degree and recommends its usage as a key role for the controller design.

At this point we would like to classify and review these bang-bang experiments in a historical context. Following the

textbook approach in classical system identification we did a lot of identification flights in the years 2005 and 2006 using separate batch runs in order to characterize the steering behavior of our kites at the various operating points. The experiments were quite cumbersome: perturbations from wind gusts can be comparatively large compared to periodic excitations caused by the deflection commands and the air path velocities were difficult to tune for the different batch runs and desired operating points. As we had to evaluate data from different days with changing environmental conditions and drifting flight properties of our kites we solely could suspect the validity of law (7). But once we switched to a bang-bang flight strategy the real law shows up clearly. This is obvious as one bang-bang experiment varies the parameter v_a over a whole range, in one flight alone, lasting only a few tens of seconds and therefore plays a trick on perturbations.

We would like to conclude this section by giving an extended version of (7) which also takes into account the effect of the gravitational force on the turn rate and reads:

$$\dot{\psi}_m = g v_a \delta + M \frac{\cos \theta_g \sin \psi_g}{v_a}. \quad (9)$$

The quantity θ_g denotes the angle between \vec{e}_{yaw} and the $\vec{e}_x \perp \vec{e}_y$ -plane and ψ_g the angle between \vec{e}_{roll} and the $\vec{e}_x \perp \vec{e}_y$ -plane.

Because a steering deflection could be regarded as a kite force component into pitch direction \vec{e}_{pitch} subsequently leading to a yaw rate, the gravity force, projected onto the pitch axis by $\cos \theta_g \sin \psi_g$, should have the same effect. We have shown in this section that the yaw rate is proportional to $v_a \delta$. This can be attributed to a side force proportional to $v_a^2 \delta$ from aerodynamical and design considerations. Transferring this reasoning to the mass term, which is independent from v_a , we expect a factor of M/v_a between the 'gravitational' side force and the yaw rate. The constant M includes system masses and kite characteristics. As M is positive for our kites we get an instable behaviour and thus have to stabilize ψ_m by active control.

As for the usual operation point of dynamic flight we have $(g v_a) \gg (M/v_a)$ and thus the second term of (9), which we call 'mass term', can be neglected for the design of the linear feedback law only to be introduced as correction term via a feedforward path to the controller structure.

As our operational flights under autopilot conditions (during the dynamic flight modes) utilize similar bang-bang like commands, we can use the discussed identification scheme as a standard tool to establish or check the kite parameter during normal operation. During flights a recursive least-square algorithm [26] runs in order to determine the system parameters g and M on-line. We monitor these values for changes which may indicate upcoming material failures in advance and can adopt controller parameters on-line, if necessary, to increase robustness.

After we have convinced ourselves of the validity of (7), we will now present the controller design which is strongly influenced in its structure by the discussed law.

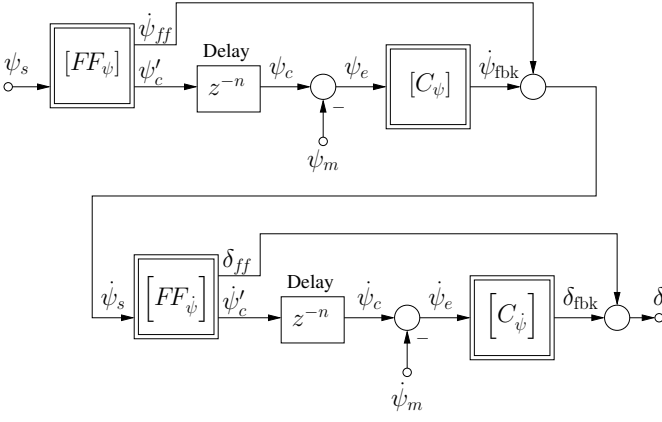


Fig. 8. Cascaded controller approach for ψ control implementing the model following structure. Detailed diagrams for the blocks $[FF_\psi]$, $[C_\psi]$, $[FF_{\dot{\psi}}]$ and $[C_{\dot{\psi}}]$ are shown in figs. 13, 12, 9 and 10. The Controller calculates a steering command δ from the input value ψ_s using a measured yaw angle ψ_m and yaw rate $\dot{\psi}_m$.

V. CONTROLLER DESIGN

Most of time the kite is operated in a highly dynamic regime where the air path speed can easily vary by a factor of up to 3–5 within some seconds and the deflection command can change by more than $\pm 60\%$. The classical way to approach such a controller design task would be to use a controller structure which specifically aims at time varying, non-linear systems. Non-linear dynamic inversion or non-linear model predictive control could be such candidates. Gain scheduling based on linearized plant models along the trajectory is a further alternative. Actually we tried the latter approach in the beginning but were not satisfied with the achievable robustness. The core of the problem is governed by the fact that it is difficult to execute a classical modelling approach as usually performed in aerospace application. Such an approach would be based on an aerodynamic database covering the full dynamic regime. Performing wind tunnel tests would be expensive and for our larger kites with 160 m^2 area downright impossible. We also learned that a controller structure where the kite trajectory is given as a set-point directly into a unique controller block, which then directly computes the deflection command, lacks robustness due to the above mentioned modelling issue.

Instead we settled for a separation of the overall controller task into 'guidance' and 'control'. Section VI describes how the guidance algorithm computes ψ_s , which is then the input to the controller. However the major distinguishing feature of our autopilot, compared to other approaches we found in the cited literature, is the cascaded controller structure which is based on the model following principle as shown in Fig. 8.

The basic idea is to reflect the separation of dynamics (7) and kinematics, as given by $\dot{\psi} = d\psi/dt$, adequately in the controller structure. We can see two cascaded loops in Fig. 8. The inner loop gets a commanded rate $\dot{\psi}_s$ as an input and computes the deflection command δ . The outer loop has ψ_s as input and commands $\dot{\psi}_s$ to the inner loop.

From (7) we recognize that the dynamics from deflection to yaw rate can be viewed as a proportional plant, $\dot{\psi}_m = K_\psi \delta$, where $K_\psi = g v_a$ denotes the gain. Of course we

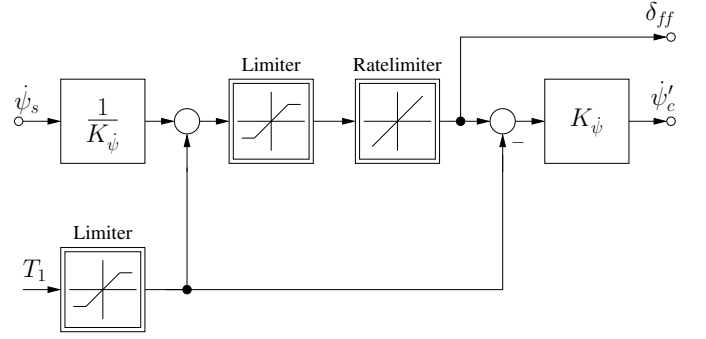


Fig. 9. Details of block $[FF_\psi]$ of the cascaded controller (see Fig. 8). From input value $\dot{\psi}_s$ the feedforward value δ_{ff} and $\dot{\psi}'_c$ are calculated by using a steering pod model consisting of a limiter and a ratelimiter. Note the translation of rate to command and back via division and multiplication by K_ψ .

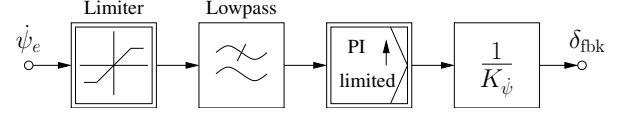


Fig. 10. Feedback block $[C_\psi]$ of the cascaded controller (see Fig. 8). The controller mainly consists of a PI feedback on the yaw rate. Note the division by K_ψ , which is a function of the air path speed, thus introducing a nonlinearity into the feedback by transforming the rate command from the PI controller to the command portion δ_{fbk} which is routed to the steering pod.

have to realize that K_ψ is not constant but a function of the air path velocity. We take care of this by employing a feedforward/feedback structure which implements the model following principle: in the feedforward term $[FF_\psi]$ we compute, in an open loop fashion, the deflection command δ_{ff} necessary to achieve $\dot{\psi}_s$. The feedback control $[C_\psi]$ only acts when there is a remaining control error due to external disturbances or due to unmodeled plant dynamics. An appropriate delay block is necessary in order to capture all the delays from command to actual execution.

With such a feedforward/feedback structure we can accommodate the dependence of the gain K_ψ from the air path velocity. Fig. 9 provides details of the feedforward controller block. In line with the idea of the model following principle the feedforward block is not limited to linear equations and can accommodate any system description. The extended version of the turn rate law (9)

$$\dot{\psi}_m = K_\psi \delta + M \frac{\cos \theta_g \sin \psi_g}{v_a} \quad (10)$$

can easily be considered. The principal idea is to invert it in order to compute the necessary deflection δ_{ff} . In addition the block also contains nonlinear elements, like limiters on angle and rate, in order to capture limited pod steering speed and other constraints. The mass term from (10) is introduced via

$$T_1 = \frac{M}{K_\psi} \frac{\cos \theta_g \sin \psi_g}{v_a}. \quad (11)$$

Fig. 10 provides the details of the feedback controller block: As the plant has a proportional character the feedback structure is of proportional/integrator (PI controller) type. The output of

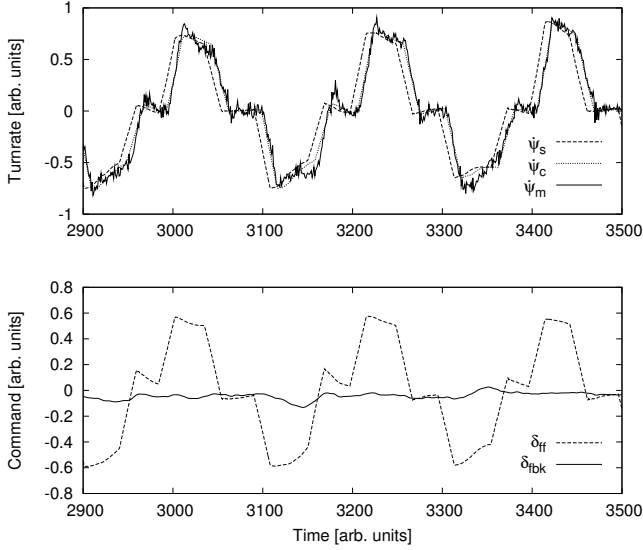


Fig. 11. Experimental results for the inner loop. In the upper plot the measured yaw rate ψ_m is compared to ψ_s and ψ_c . The lower plot compares feedforward δ_{ff} and feedback δ_{fbk} controller output signals for a typical dynamical flight situation.

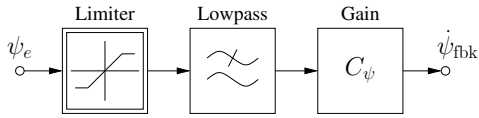


Fig. 12. Feedback block $[C_\psi]$ of the cascaded controller (compare Fig. 8). As the dynamics to be controlled is mainly an integrator, a proportional feedback has been chosen.

the PI controller is divided by K_ψ , thus taking the velocity dependence into account. A lowpass is added for noise rejection of the measurement.

By providing Fig. 11 we illustrate how effective the inner loop actually works. We show how a couple of repeating flight patterns look at the inner loop level. The feedforward command δ_{ff} moves between $\pm 60\%$ of the available deflection range as can be seen in the lower graph of the figure. The feedback command δ_{fbk} hardly needs to correct control errors due to unmodeled plant dynamics. Actually this figure is just an alternative account to Fig. 6 in proving the good fit of the turn rate law.

As seen from the outer loop the inner loop has dealt with the aerodynamically influenced part of the dynamics. It now remains for the outer loop controller to achieve a desired ψ_s . The division into feedforward and feedback parts is kept. Figs. 12 and 13 provide further details.

As the plant characteristic is of integrating nature a feed-back law with proportional character $[C_\psi]$, augmented by a lowpass, is sufficient. As before a limiter is also employed. The feedforward block $[FF_\psi]$ (see Fig. 13) has more elaborate features. Note that the feedforward term features an internal feedback loop. The basic idea is to shape the commanded ψ_s , even if it is a jump, in such a way that it corresponds to the actual response capabilities of the control pod and kite. We achieve this by employing an internal loop from ψ_s to ψ'_c . Note that the two limiters are crucial in shaping the final ψ'_c

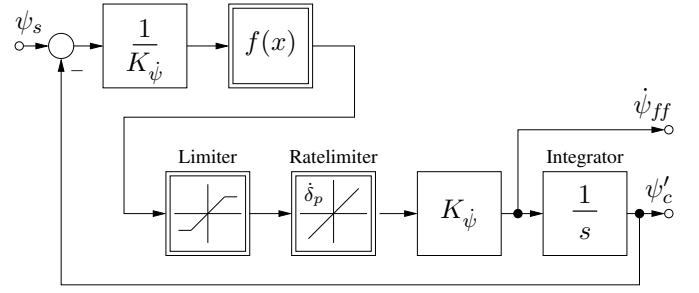


Fig. 13. Feedforward block $[FF_\psi]$ details of cascaded controller (see Fig. 8). The steering pod model is included as combination of limiter and ratelimiter. Note that even for step inputs on ψ_s the algorithm computes feedforward ψ_{ff} and ψ_c in a way which is consistent with the capability of the overall dynamics. This is achieved by using an inner feedback loop embedding the steering pod model.

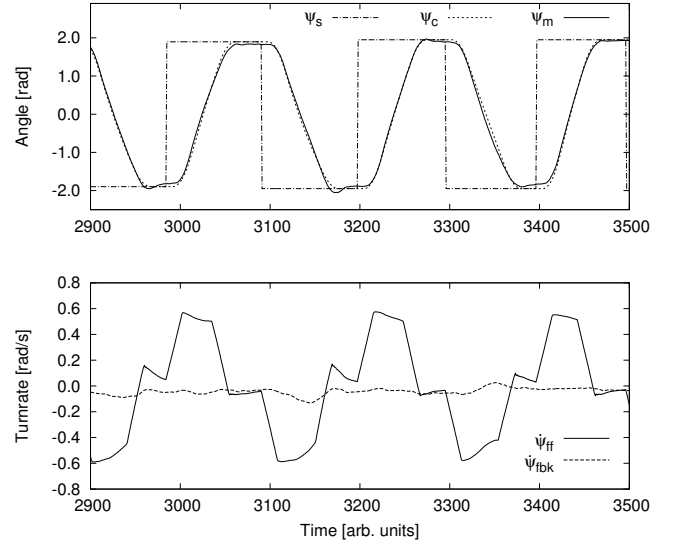


Fig. 14. Flight results for the outer loop. The upper plot compares the response of the measured angle ψ_m to a given rectangular ψ_s . The curve ψ_c is computed by the internal loop of the feedforward block $[FF_\psi]$ using a model of the steering pod (compare Fig. 13).

evolution. Furthermore we have a nonlinear function inside the loop: $f(x) = \sqrt{2\delta_p x}$. We will stop short in deriving the details of this special feature. Instead we will illustrate it with Fig. 14. The upper part shows how the input command ψ_s is shaped into the command ψ_c by the non-linear feature of the internal feedback loop. ψ_c corresponds to an actually flyable ψ_c pattern. ψ_{ff} shows the corresponding necessary rate pattern which is fed into the inner loop.

As a conclusion we will summarize the basic design principles of our controller. The first feature is the separation of the dynamics of deflection to rate (7) from the kinematic of rate to angle. This separation allows to introduce non-linear elements (mainly limiters) at the appropriate places. That way we achieve a shaping of our commanded signals such that they correspond to the limitations of the complete chain from software command over control pod steering to kite movement.

A second characteristic feature is the feedforward/feed-back separation which allows to decouple non-linear elements, as for example the mass term, from the feedback. The feed-

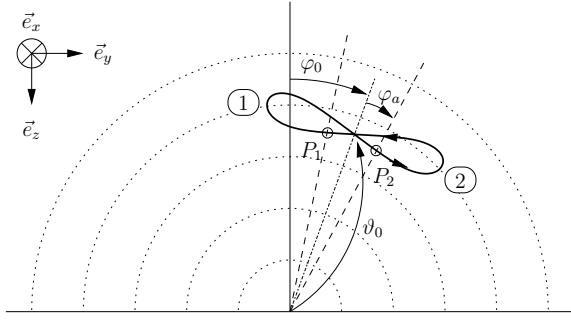


Fig. 15. Geometry for pattern generation. The figure-eight pattern is guided by two states (1) and (2). Transitions between those are triggered by the conditions $P_1: \varphi < \varphi_0 - \varphi_a$ and $P_2: \varphi > \varphi_0 - \varphi_a$. The corresponding sequence is shown in Fig. 16.

back loops can then be designed within the realm of linear control theory. Only proportional or integral dynamics remain which can be handled in a classical way. Due to the feedforward/feedback separation the selection of the closed loop bandwidth of the two loops is more concerned with achieving sufficient stability margin than with achieving performance in terms of fast response because this is already mastered by the feedforward.

The next chapter will illustrate the generation of the ψ_s command from the desired kite trajectory. Although it could be perceived as just a further cascaded loop we treat it more like the 'guidance' feature of classical aerospace applications.

VI. PATTERN GENERATION

In this section we describe the dynamic flight mode which is used to generate traction force by flying dynamical patterns in order to obtain high air path speeds and forces. The algorithm utilizes the presented controller design by providing the input value ψ_s .

In order to explain the main principles we would like to review the space station experiment of section II. In this model a constant value of $\psi_s = +\psi_0$ or $-\psi_0$ leads to a circular orbit clockwise or counterclockwise dependent on the sign of ψ_s and the obtained force can be easily controlled by the value of $|\psi_s|$. For the purpose of line force generation in our space station experiment this would finish our design effort — but in our application we are not able to fly circular orbits as the kite would crash onto the water surface. Thus the solution is to turn around at certain points of the orbit and fly back and forth.

The resulting trajectory of such a scheme is shown in Fig. 15. The underlying algorithm is similar to those of the bang-bang experiments in section IV. A constant value $\psi_s = +\psi_0$ is commanded until point 'P₁' is reached (at $\varphi \leq \varphi_0 - \varphi_a$) triggering the command $\psi_s = -\psi_0$ until point 'P₂', then triggering the former value $\psi_s = +\psi_0$ and so on. The timing of ψ_s is depicted in Fig. 16.

It is acceptable to apply a square signal to ψ_s as the controller design contains an internal model, which leads to a calculation of ψ_c which is feasible to fly. We would like to mention that every curve involves a decrease of air path speed as ϑ increases and hence a decrease of traction force. As a

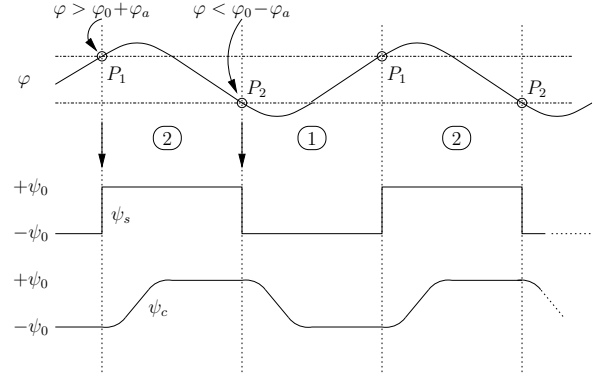


Fig. 16. Pattern generation sequence toggling between the two states (1) and (2) when conditions $P_1: \varphi < \varphi_0 - \varphi_a$ and $P_2: \varphi > \varphi_0 - \varphi_a$ are met respectively. The states directly result in a square signal to ψ_s which is shaped by the feedforward block $[FF_\psi]$ into ψ_c (compare sec. V). The respective pattern geometry is displayed in Fig. 15.

consequence curves should be commanded in an efficient way using appropriate deflections and manageable steering speeds. This is fulfilled by the controller design as it both utilizes the given curve deflection capability and the given design speed of the control pod for step transitions (see section V).

As illustrated in Fig. 15 there are three parameters defining the trajectory. The parameter φ_a determines the pattern size; the parameters φ_0 and ϑ_0 determine the center point of the pattern. The value φ_0 can be freely chosen within a certain range by the operator or an overlying algorithm in order to optimize the force component pointing in forward direction of the vessel. The value ϑ_0 can not be tuned directly but indirectly by the ψ_0 value which is the tuning knob for the air path speed v_a and hence force. Nevertheless (6) does not hold in its simple way and could be improved by using $\langle \psi \rangle = \int dt |\psi(t)|$ instead of ψ_0 in order to estimate ϑ or resulting forces.

As it is cumbersome to predict the exact wind situation at flight altitude in any case we make use of another approach for force control: we use an outer loop force controller evaluating the height of the force peaks while flying the figure-eight pattern which provides a feedback value for ψ_0 . Details of this control law as well as of the supervision mechanisms during operative flight and start procedure of the pattern are important and interesting issues each but would exceed the purpose of this paper. In Fig. 17 we present the trajectory of some eights and show the corresponding time series of the angles φ_m , ψ_s , ψ_c as well as of the air path and wind speeds.

For sake of completeness we would like to note control strategies for the neutral flight mode introduced in section II. The determining equation is (4) and the issue is to control φ to the given set value φ_0 . We mainly use a linear controller based on a classical PID controller in order to compute ψ_s from $\varphi_m - \varphi_0$.

VII. CONCLUDING REMARKS

Dealing with a complex system in a demanding environment we could present several further topics on our control system which are closely related to the discussed contents. These topics have been omitted in the previous sections because

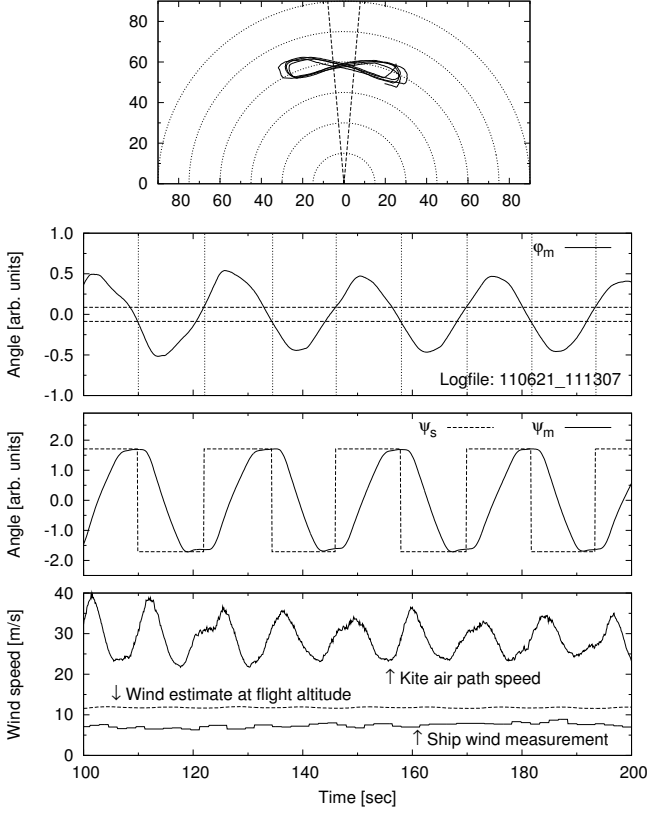


Fig. 17. Flight test results illustrating pattern generation using a 160 m² kite at an operational towing line length of 300 m. The trajectory for a typical figure-eight pattern is shown in the upper plot (compare Fig. 15). The two plots in the middle show corresponding curves for φ_m , ψ_c and ψ_m (see Fig. 16). The lower data plot shows the measured wind speeds. Note that the dynamical flight mode leads to an air path speed of factor 3–4 higher than the estimated wind speed at flight altitude which is significantly higher than the wind speed measured aboard the ship.

we wanted a clear outline of the major features of the algorithm. They are now briefly summarized to give a more comprehensive understanding.

First of all we have completely excluded the ship from our treatment by arguing that consideration of the apparent wind, which is the wind speed with respect to the ship's motional frame of reference, is an appropriate approximation for the basic design of the control system. Furthermore the choice of pattern parameters [25], the optimization of the towing force with respect to the ship forward direction [27] as well as influences on dynamics due to waves are improvements to an operational kite autopilot but have not been presented here.

It is worth mentioning that both, the thorough choice of the sensor set up and preprocessing algorithms, contribute a significant and crucial part to the presented controller performance and robustness. An important task is the estimation of the angle ψ which involves not only sophisticated inertial navigation algorithms but also takes into account estimates for wind speed and wind direction at flight altitude as these may vary on the timescale of minutes. A further discussion of these mainly technical and cumbersome topics will be subject to future publications.

In our discussions we have made use of constant and long

line lengths which is the common operating point of our system. Winching of the towing line is done while launching and recovering our system and goes along with extra dynamic effects which are considered as additional correction terms to the presented equations. Especially for control during launch and recovery at shorter line lengths we use the idea of receding horizon feedback from the predictive control (NMPC) approaches [28] in order to provide ψ_s . However our optimization computation is done analytically (thanks to the special and simple structure of the design model, see III) as opposed to having a numerical solver.

Wind gusts may also go along with downwinds leading to a free flight situation of the usually constrained system. This situation involves a completely different dynamics. While (7) still holds in large part the kinematic changes significantly. It is a crucial advantage of the particular choice of coordinate system (see section II) that controller input values behave in a favorable way during these exceptional occurrences. We would like to note that also disturbances due to excitations of internal modes of the real system are effectively suppressed although these modes are not considered explicitly in the 'design model'.

A more elaborate challenge is the avoidance of and response to stall situations. Systematic experimental tests on this are hardly feasible. Nevertheless it is an important issue and subject to current research and development activities.

VIII. SUMMARY

In this paper we have presented a simple dynamical model for the dynamics of constrained kites. A key point has been that for controller design the complete aerodynamics can be reduced to a law involving only one or two quantities (see (7) and (9), respectively). We have discussed flight data to justify that this reduction describes reality to a surprising high degree. Utilizing this we have developed a cascaded controller, based on the model following principle approach, and proved the effectiveness with flight test data. Finally the main principles of flight pattern generation have been given.

We would like to finish this paper by emphasizing that the presented results hold for the SkySails towing kite system but the underlying equations are commonly valid for the dynamics of constrained kites. Thus a lot more applications of the presented controller are feasible especially in the fascinating upcoming field using kites in order to open up the green resource of wind power.

APPENDIX DERIVATION OF SYSTEM DYNAMICS

The system vectors read (see Fig. 2):

$$\vec{e}_{\text{roll}} = \begin{pmatrix} -\sin \vartheta \cos \psi \\ -\cos \varphi \sin \psi + \sin \varphi \cos \vartheta \cos \psi \\ -\sin \varphi \sin \psi - \cos \varphi \cos \vartheta \cos \psi \end{pmatrix} \quad (12)$$

$$\vec{e}_{\text{pitch}} = \begin{pmatrix} \sin \vartheta \sin \psi \\ -\cos \varphi \cos \psi - \sin \varphi \cos \vartheta \sin \psi \\ -\sin \varphi \cos \psi + \cos \varphi \cos \vartheta \sin \psi \end{pmatrix} \quad (13)$$

$$\vec{e}_{yaw} = \begin{pmatrix} -\cos \vartheta \\ -\sin \varphi \sin \vartheta \\ \cos \varphi \sin \vartheta \end{pmatrix}. \quad (14)$$

As discussed in section II we neglect gravity. Thus the problem becomes independent of φ and can be treated for $\varphi = 0$ without loss of generality which implies the following basis vectors:

$$\vec{e}_{roll} = \begin{pmatrix} -\sin \vartheta \cos \psi \\ -\sin \psi \\ -\cos \vartheta \cos \psi \end{pmatrix} \quad (15)$$

$$\vec{e}_{pitch} = \begin{pmatrix} \sin \vartheta \sin \psi \\ -\cos \psi \\ \cos \vartheta \sin \psi \end{pmatrix} \quad (16)$$

$$\vec{e}_{yaw} = \begin{pmatrix} -\cos \vartheta \\ 0 \\ \sin \vartheta \end{pmatrix}. \quad (17)$$

The air flow \vec{v}_a of the flying system is

$$\vec{v}_a = \begin{pmatrix} v_0 \\ 0 \\ 0 \end{pmatrix} - v_r \vec{e}_{roll} - v_p \vec{e}_{pitch}. \quad (18)$$

The first term describes the external wind, the subsequent two terms the flow due to the kinematic speeds v_r and v_p with respect to the basis vectors \vec{e}_{roll} and \vec{e}_{pitch} .

Considering the basic aerodynamics of an airfoil [22] as a very simple model we claim validity of the following two conditions:

- 1) The airflow vector lies in the $\vec{e}_{roll} \perp \vec{e}_{yaw}$ -plane which means

$$(\vec{e}_{pitch}, \vec{v}_a) = 0. \quad (19)$$

- 2) The airflow direction with respect to \vec{e}_{roll} and \vec{e}_{yaw} is given by the glide ratio E which is the ratio between lift and drag coefficients $E = L/D$,

$$\frac{(\vec{e}_{roll}, \vec{v}_a)}{(\vec{e}_{yaw}, \vec{v}_a)} = E. \quad (20)$$

Insertion of the definitions (15)-(17) into (19) and (20) yields the velocity components

$$\frac{v_p}{v_0} = \sin \vartheta \sin \psi \quad (21)$$

$$\frac{v_r}{v_0} = E \cos \vartheta - \sin \vartheta \cos \psi. \quad (22)$$

The airflow in roll direction v_a , measured by the anemometer in the control pod, can be calculated using (15) and (18) by:

$$v_a = -(\vec{v}_a, \vec{e}_{roll}) = v_0 E \cos \vartheta. \quad (23)$$

Geometric and kinematic considerations lead to the relation

$$\dot{\vartheta} = \frac{1}{L} (v_r \cos \psi - v_p \sin \psi). \quad (24)$$

Using (21), (22) and (23) we get for the dynamics of ϑ the following differential equation:

$$\dot{\vartheta} = \frac{v_a}{L} \left(\cos \psi - \frac{\tan \vartheta}{E} \right). \quad (25)$$

Similarly we obtain for φ the equation

$$\dot{\varphi} = \frac{1}{L \sin \vartheta} (-v_r \sin \psi - v_p \cos \psi), \quad (26)$$

and using (21), (22) and (23) the equation of motion

$$\dot{\varphi} = -\frac{v_a}{L \sin \vartheta} \sin \psi. \quad (27)$$

For constant values of ψ one obtains the steady-state solution of (25) as

$$\vartheta_0(\psi) = \arctan(E \cos \psi). \quad (28)$$

ACKNOWLEDGEMENTS

We acknowledge support by the whole SkySails team, especially contributions from the kite, software, hardware and mechanical development teams. Their knowledge and excellent work on the system as well as the unfatiguing support by the test engineers and nautical crews during numerous sea trials are crucial contributions to the findings presented here.

REFERENCES

- [1] SkySails GmbH. [Online]. Available: <http://www.skysails.de>
- [2] M. Canale, L. Fagiano, and M. Milanese, "Power kites for wind energy generation," *IEEE Control Syst. Mag.*, pp. 25–38, Dec. 2007.
- [3] M. Canale, L. Fagiano, M. Milanese, and M. Ippolito, "Kitegen project: control as key technology for a quantum leap in wind energy generators," *Proc. of American Control Conference, New York*, Jul. 2007.
- [4] L. Fagiano, M. Milanese, and D. Piga, "High altitude wind power generation," *IEEE Trans. Energy Convers.*, vol. 25, no. 1, pp. 168–180, 2010.
- [5] P. Williams, B. Lansdorp, R. Ruiterkamp, and W. Ockels, "Modeling, simulation, and testing of surf kites for power generation," *Proc. Modelling and Simulation Technologies Conf. AIAA 2008-6693*, Aug. 2008.
- [6] KITEng Altitude Wind Generation. [Online]. Available: <http://www.kitenrgy.net>
- [7] EnerKite GmbH. [Online]. Available: <http://www.enerkite.de>
- [8] SwissKitePower collaborative Research and Development Project. [Online]. Available: <http://www.swisskitepower.ch>
- [9] Makani Power Inc. [Online]. Available: <http://www.makanipower.com>
- [10] Ampyx Power. [Online]. Available: <http://www.ampyxpower.com>
- [11] "Book of abstracts," ISBN 978-94-6018-370-6, Airborne Windenergy Conference 2011. [Online]. Available: <http://www.awec2011.com>
- [12] A. Ilzhöfer, B. Houska, and M. Diehl, "Nonlinear MPC of kites under varying wind conditions for a new class of large scale wind power generators," *Int. J. Robust Nonlinear Control*, vol. 17, no. 17, pp. 1590–1599, Nov. 2007.
- [13] P. Williams, B. Lansdorp, and W. Ockels, "Optimal crosswind towing and power generation with tethered kites," *AIAA Journal of Guidance, Control and Dynamics*, vol. 31, no. 1, pp. 81–93, Jan. 2008.
- [14] —, "Nonlinear control and estimation of a tethered kite in changing wind conditions," *AIAA Journal of Guidance, Control and Dynamics*, vol. 31, no. 3, pp. 793–799, 2008.
- [15] A. Furey and I. Harvey, "Evolution of neural networks for active control of tethered airfoils," *Proc. The 9th European Conference on Artificial Life*, pp. 746–756, 2007.
- [16] L. Fagiano, "Control of tethered airfoils for high-altitude wind energy generation," Ph.D. dissertation, Politecnico di Torino, Italy, 2009.
- [17] B. Houska and M. Diehl, "Robustness and stability optimization of power generating kite systems in a periodic pumping mode," *Proc. of the IEEE Multi-Conference on Systems and Control*, Sep. 2010.

- [18] J. H. Baayen and W. J. Ockels, "Tracking control with adaption of kites," *IET Control Theory and Applications*, vol. 6, no. 2, pp. 182–191, 2012.
- [19] G. Dadd, "Development, validation, and demonstration of a test rig for kite performance," Master's thesis, University of Southampton, GB, 2001.
- [20] B. Lansdorp, R. Ruitkamp, and W. Ockels, "Towards flight testing of remotely controlled surfkites for wind energy generation," *Proc. Modelling and Simulation Technologies Conf. AIAA-2007-6643*, Aug. 2007.
- [21] M. Canale, L. Fagiano, and M. Milanese, "High altitude wind energy generation using controlled power kites," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 279–293, 2010.
- [22] J. S. Lingard, "The aerodynamics of gliding parachutes," *Proc. 9th Aerodynamics Decelerator and Ballon Tech. Conf., AIAA-86-2427-CP*, Oct. 1986.
- [23] P. Williams, B. Lansdorp, and W. Ockels, "Flexible tethered kite with moveable attachment points, part I: Dynamics and control," *Proc. Atmospheric Flight Mechanics Conf. AIAA-2007-6628*, Aug. 2007.
- [24] M. Diehl, "Real-time optimization for large scale nonlinear processes," Ph.D. dissertation, University of Heidelberg, Germany, 2001.
- [25] G. M. Dadd, D. A. Hudson, and R. A. Sheno, "Determination of kite forces using three-dimensional flight trajectories for ship propulsion," *Renewable Energy*, vol. 36, pp. 2667–2678, 2011.
- [26] L. Ljung, *System Identification*, 2nd ed. Prentice Hall PTR, 2007.
- [27] B. Houska and M. Diehl, "Optimal control of towing kites," *Proc. IEEE Conference on Decision and Control*, pp. 2693–2697, 2006.
- [28] J. M. Maciejowski, *Predictive Control - with Constraints*, 1st ed. Prentice Hall PTR, 2002.

Michael Erhard received the diploma degree in physics (2000) from University of Freiburg, Germany and the Ph.D. degree in physics (2004) from University of Hamburg, Germany.

His main research interests have been experiments in quantum optics and laser physics but also theoretical quantum optics. For the Ph.D. degree he worked on experiments with multi-component Bose–Einstein condensates.

Dr. Erhard joined SkySails end of 2004 as development engineer responsible for the flight control system (autopilot). Since then he has been deeply involved in several hardware and software designs on the sensor data acquisition system and the kite steering units. His main responsibility is the development of sensor data processing and control algorithms for the flying system as well as evaluation of those in flight tests.

Hans Strauch received the diploma degree in physics (1981) from University of Kiel, Germany. Afterwards he joined Anschütz GmbH, Kiel, developing autopilots and ground track controllers for ships. In 1988 he moved to Astrium Space Transportation, Bremen, Germany where he developed the guidance and control algorithms for various vehicles ranging from upper stages of launchers to re-entry bodies. He currently holds the position of Senior Expert GNC.

From 1998 until 2002 he developed at NASA, Johnson Space Center, Texas, USA the guidance and control software for the parafoil phase of the X38 Crew Rescue Vehicle. In 2004 he was responsible for the attitude control of the winged body, autonomous landing demonstrator PHOENIX.

Mr. Strauch has been working as a consultant to SkySails since beginning of 2004. His contributions to the findings reported in this paper are independent from his affiliation to Astrium.